

Network Automata: Coupling structure and function in real-world networks

David M.D. Smith^{1,2,3,4,*}, Jukka-Pekka Onnela^{2,3,5}, Chiu Fan Lee², Mark Fricker⁶, and Neil F. Johnson⁷

¹*Centre for Mathematical Biology, Oxford University, Oxford OX1 3LB, U.K.*

²*Department of Physics, Clarendon Laboratory, Oxford University, Oxford OX1 3PU, U.K.*

³*CABDyN Complexity Centre, Oxford University, Oxford, OX1 1HP, U.K.*

⁴*Oxford Centre for Integrative Systems Biology, Department of Biochemistry, Oxford University, South Parks Road, Oxford, OX1 3QU, U.K.*

⁵*DBEC, Helsinki, University of Technology, P.O. Box 9203, FIN-02015 HUT, Finland.*

⁶*Department of Plant Sciences, University of Oxford, Oxford, OX1 3RB, U.K. and*

⁷*Physics Department, University of Miami, Coral Gables, Florida FL 33124, USA.*

(Dated: July 17, 2009)

We introduce Network Automata, a framework which couples the topological evolution of a network to its structure. It is useful for dealing with networks in which the topology evolves according to some specified microscopic rules and, simultaneously, there is a dynamic process taking place on the network that both depends on its structure but is also capable of modifying it. It is a generic framework for modeling systems in which network structure, dynamics, and function are interrelated. At the practical level, this framework allows for easy implementation of the microscopic rules involved in such systems. To demonstrate the approach, we develop a class of simple biologically inspired models of fungal growth.

PACS numbers: 89.75.Fb, 87.15.A-, 87.85.Xd

I. INTRODUCTION

The network framework has proved very successful in the study of various complex interacting systems. In the network description, the interacting elements are depicted as nodes and the interactions between the elements are represented by links connecting the corresponding nodes. The science of complex networks has progressed very quickly in the last few years, and some excellent reviews have been written covering both the methodology and key results [1, 2, 3]. The strength of the complex network paradigm lies in its ability to capture some of the essential structural characteristics of interacting systems while reducing the details of both the elements and their interactions. Consequently, the early complex network literature was almost exclusively focused on structural properties of networks.

In many out-of-equilibrium growing networks, the evolution at a given time is dependent on the configuration of the network at that time, as exemplified in the preferential attachment model of Barabási and Albert [4]. In this paper, we develop the Network Automata (NA) framework, which can be seen as a natural extension of the Cellular Automata (CA) framework [5]. When employed to model dynamic, functional networks, the NA framework not only removes ambiguity at an implementation level through exhaustive specification of the microscopic ruleset employed, it also provides a platform for comparison between apparently different network algorithms.

While structural properties remain important in constraining the behavior of the system, there is also sig-

nificant interest in understanding dynamical processes taking place on networks [6]. Indeed, the marriage between structure and dynamic processes is so strong that the behavior of dynamic processes can be used to detect structure [7]. While a network's topology constrains the type of dynamics that may unfold on it, in many scenarios the dynamical process may influence the subsequent evolution of the topology - meaning that the structural properties of the network are coupled to its function. A real-world example might be the evolution of transport links within a city. The dynamics of the human population using this network in turn affect the reinforcement or removal of the those transport links and the feedback process is apparent. A similar situation may arise in the context of social networks, where one's current social opportunities and dynamics are limited by the existing network structure, but they can be widened by extending the network.

There have been several specific attempts in the literature to inter-relate a network's structure, dynamics, and function [8, 9, 10, 11]. However, the present paper distinguishes itself by providing a *generic* framework for dealing with these types of systems. While many network generating algorithms and Cellular Automata models can be easily reproduced within the Network Automata framework [12], its usefulness is exemplified when it is applied to systems in which topological evolution is coupled to a dynamic process occurring upon the network. Its contribution to the network field lies in the fact that it enables a *precise* specification of the microscopic rules underlying the structural *and* functional evolution of a given network-based system, thereby enabling comparison between different classes of network. We describe the framework in Section II and as a demonstration, we introduce a class of progressively more realistic, biologically-inspired models of woodland fungal growth in Section III.

*Electronic address: d.smith3@physics.ox.ac.uk

Although the models consist of simple rulesets, they are nevertheless capable of producing structures qualitatively similar to mycelial network development (e.g. Fig. 1). This example is *Phanerochaete velutina*, a foraging saprotrophic woodland fungus that builds an adaptive network to translocate resource [13, 14, 15]. The transport flux is oscillatory and still not fully understood [14, 15, 16, 17].

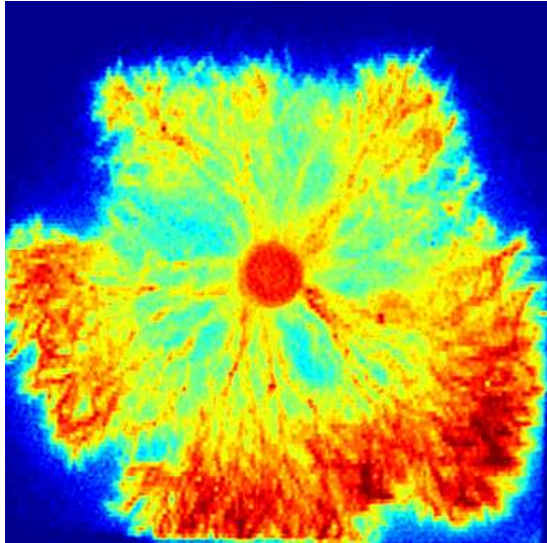


FIG. 1: (Color online) Transport of non-metabolised, radioactively labelled amino acid within *Phanerochaete velutina*, a foraging woodland fungi. The image was taken using photon counting scintillation imaging [14, 15].

II. NETWORK AUTOMATA

Consider an arbitrary weighted or unweighted, directed or undirected network at some time t which is to be grown to some size N_{tot} . As nodes might be added to the system at each time step, the network would conventionally be considered to be growing. An alternative representation is to treat the system as being of size N_{tot} at all times where at some time t many of the nodes have no links. Information regarding the network's topology is entirely encompassed within the adjacency matrix $\mathbf{A}(t)$ which is of dimension $N_{tot} \times N_{tot}$. The matrix holds information about which links exist, their direction and, perhaps, weights. One might consider the evolution of the network as a process that alters the elements within this adjacency matrix, updating the attributes of any of the possible links which could exist in the system. We base a framework of network growth around this concept. If the microscopic ruleset governing the network's evolution is solely related to quantities which can be derived from the network's current topology (and hence from $\mathbf{A}(t)$) then the evolution of the network might be expressed in terms of some operation F acting upon the adjacency matrix:

$$\mathbf{A}(t+1) = F(\mathbf{A}(t)). \quad (1)$$

The ruleset employed could relate to any property of the nodes (their degree, betweenness, clustering and so on) or the links (weights or direction) and might be deterministic or stochastic.

This concept can be achieved in practice by visiting *all* possible links within the adjacency matrix $\mathbf{A}(t)$ for a network comprising N_{tot} nodes (whether part of a component or not) once every time step. The update as to the nature of the link at the next time step (its existence or its weight or direction) is then prescribed by the ruleset [21]. The ruleset can take the form of a lookup table in which the *state* of a link is evaluated and the *color* (existence/direction/weight) is prescribed for the next timestep. This update process is analogous to the update of a cell within a conventional Cellular Automaton [18] *except* that it acts upon the connectivity (or neighbourhood) of an node, thereby generating $\mathbf{A}(t+1)$. This link-orientated update is a generic description of a dynamic network and all the essential features of that network's evolution are then contained within the exhaustive ruleset. We describe such a system as a *Network Automaton* [22]. No restriction has yet been made as to the directionality or weight of links.

Now consider a situation in which the topology of a network evolves while there is simultaneously some process taking place on the network. At any given time the topology of the network constrains the type of dynamics that may unfold on it. However, the dynamical process may influence the subsequent topological evolution of the network, so that its structural properties are coupled to its function and vice versa. The ruleset governing the topological update process relates not only to network related quantities but also functional aspects of the nodes and/or links. Since the functional process requires a network on which to perform, we decouple the evolution of the network into two distinct phases, namely, that affecting its topology and that governing the functional process. Writing the functional information (relating to nodes and/or links) at some time t as some matrix $\mathbf{S}(t)$, the formal description of the evolution can be expressed in terms of some operations F and G as

$$\mathbf{A}(t+1) = F(\mathbf{A}(t), \mathbf{S}(t)), \quad (2)$$

$$\mathbf{S}(t+1) = G(\mathbf{A}(t+1), \mathbf{S}(t)). \quad (3)$$

This expression states that the network evolves according to some process, which is determined by its own current topology $\mathbf{A}(t)$, *and also* by some attributes of its nodes and links that includes function-based information $\mathbf{S}(t)$ [23]. The functional process then occurs on this network to generate the new set of information $\mathbf{S}(t+1)$. The global state of the system is encompassed by the matrices \mathbf{A} and \mathbf{S} .

III. BIOLOGICALLY INSPIRED MODEL

We now construct a series of simple models of woodland fungi [13, 15, 16, 19] to demonstrate the versatil-

ity of the Network Automata framework. Although the model, which describes the growth of the fungi and distribution of resources within it, is biologically inspired, its aim is not to incorporate a large number of biological details. Instead, we adopt a minimalist approach to emulate fungal growth and internal nutrient transport from a small set of microscopic rules. We start by specifying biologically naive and mathematically simple update rules (Ruleset **a**) which govern the topological part of the system evolution as in Eq. 2. This is paired to a simple functional update process (Process **1**) representing Eq. 3. The two combined fully define the model. We then modify both the topological rules (Ruleset **b** and **c**) and functional process (Process **2**) according to some basic physical and biological considerations providing 6 distinct models. The end product may serve as a platform for more elaborate future models of fungal growth, demonstrating the effectiveness of using the framework.

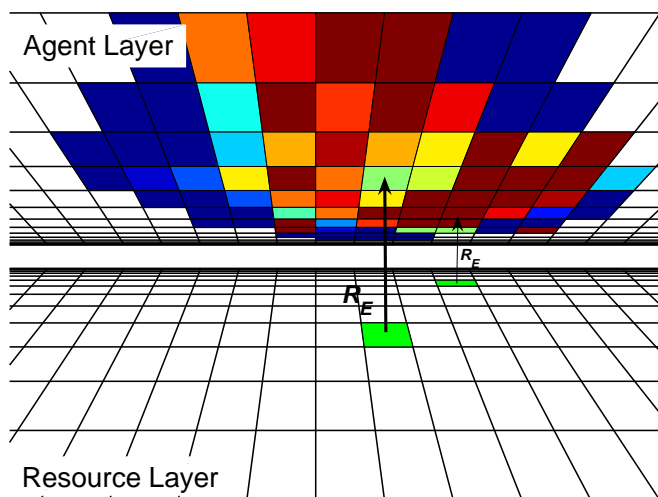


FIG. 2: (Color online) A biologically inspired multi-agent model whereby the agent layer is superimposed upon a resource layer. Agents above a resource can accumulate resources at some rate R_E .

Consider a system of agents who might each be interpreted as a cell in a two dimensional lattice. The connectivity between agents is North, South, East and West reflecting a possible connectivity of 4, so that agents are restricted to local information as long range communication is assumed unlikely in the biological system. The agent layer is superimposed on a resource layer as in Fig. 2. The rules of the system are very simple. If an agent is above a resource, it absorbs that resource at some rate R_E . The objective of each agent is two-fold: to grow into available space in the search for more resource, and to redistribute excess internal resource to support further exploration. We allow each agent to grow only one new neighbor (in a random direction) at a particular time, but only if the agent has resources to do so. To mimic active transport of resources to the growing tip, an agent passes resources to its neighbor cell provided

that the neighbor does not pass resources to it. We now endeavor to categorize this simple multi-agent system in the NA framework. This serves not only to clarify any ambiguities that arise in the programming of a multi-agent system, but also as a potential aid to improving efficiency in that the required iteration and information storage/retrieval aspects are clearly defined by the rule-sets imposed.

Let us first look at the growth (topological) stage. Each cell or agent represents a node and the boundary between two cells through which resource is passed is represented by a link. Consider that the information upon which the topological ruleset will act to update the attributes of a link in the network is simply the amount of resource that each of the two nodes has at each end of the link and their in and out-degrees. We can simply write the functional information as a vector such that $S_i(t)$ refers to the resource (functional variable) that agent (node) i has at time t . For clarity, we can express some topological information as vectors too, such that the element $k_i(t)$ represents the total degree of a node, $\text{out}k_i(t)$ its out-degree which, in turn, provides its in-degree $\text{in}k_i(t)$, each of which is obtainable from the adjacency matrix $\mathbf{A}(t)$.

We will grow this Network Automaton in an unweighted but directed adjacency matrix \mathbf{A} so that if $A_{i,j} = 1$ the link exists and is directed from i to j , whereas if $A_{j,i} = 1$ the link exists and is directed from j to i . If neither $A_{i,j} = 1$ nor $A_{j,i} = 1$ then the link does not exist. Here $A_{i,j}$ and $A_{j,i}$ are mutually exclusive. As each node has limited possible connectivity (here 4), we only consider the subset of links in the system which could possibly exist. The structural update process runs through all of these possible links and each pair of nodes which could be connected is considered once. The link attributes $A_{i,j}$ and $A_{j,i}$ are then updated at the same time.

We can now write the network update procedure for this topological update as Ruleset **a** in terms of an exhaustive truth table as in Table I. Note that if $A_{i,j}(t) = A_{j,i}(t) = 0$, the total degree of both nodes i and j is less than the connectivity d of the lattice in which the system evolves. We denote the outcome of a Bernoulli trial as $\omega(p)$ such that $P(\omega(p) = 1) = p$ and conversely $P(\omega(p) = 0) = 1 - p$. We employ a step function $\phi(x)$ defined as $\phi(x) = 1$ for $x > 0$ and $\phi(x) = 0$ for $x \leq 0$.

We now describe the resource distribution (functional) stage and start by mapping the adjacency matrix $\mathbf{A}(t+1)$ to a normalised transition matrix $\mathbf{T}(t+1)$ describing the flow of resource between adjacent cells:

$$\begin{aligned} T_{i,j}(t+1) &= \begin{cases} A_{i,j}(t+1)/\text{out}k_i(t+1) & \text{for } \text{out}k_i(t+1) > 0 \\ 0 & \text{for } \text{out}k_i(t+1) = 0 \end{cases} \\ T_{i,i}(t+1) &= \begin{cases} 0 & \text{for } \text{out}k_i(t+1) > 0 \\ 1 & \text{for } \text{out}k_i(t+1) = 0. \end{cases} \end{aligned} \quad (4)$$

Through this process, an agent distributes resource equally amongst those neighbours which are not transmitting resource to it.

Model a				Model b				Model c			
time t				time $t + 1$		time $t + 1$		time $t + 1$		time $t + 1$	
$A_{i,j}(t)$	$A_{j,i}(t)$	$\phi(S_i(t))$	$\phi(S_j(t))$	$A_{i,j}(t+1)$	$A_{j,i}(t+1)$	$A_{i,j}(t+1)$	$A_{j,i}(t+1)$	$A_{i,j}(t+1)$	$A_{j,i}(t+1)$	$A_{i,j}(t+1)$	$A_{j,i}(t+1)$
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	$\omega\left(\frac{1}{d-k_j(t)}\right)$	0	$\omega\left(\frac{1}{d-k_j(t)}\right)$	0	$\omega\left(\frac{g}{d-k_j(t)}\right)$	0	$\omega\left(\frac{g}{d-k_j(t)}\right)$
0	0	1	0	$\omega\left(\frac{1}{d-k_i(t)}\right)$	0	$\omega\left(\frac{1}{d-k_i(t)}\right)$	0	$\omega\left(\frac{g}{d-k_i(t)}\right)$	0	0	0
0	0	1	1	$\omega\left(\frac{1}{2}\right)$	$1 - A_{i,j}(t+1)$	$\omega\left(\frac{1}{2}\right)$	$1 - A_{i,j}(t+1)$	$\omega\left(\frac{1}{2}\right)$	$\omega\left(\frac{1}{2}\right)$	$1 - A_{i,j}(t+1)$	$1 - A_{i,j}(t+1)$
0	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0	1	0	1
0	1	1	0	0	1	0	1	0	1	0	1
0	1	1	1	0	1	$\delta(\text{in}k_i, d)\omega\left(\frac{1}{d}\right)$	$1 - A_{i,j}(t+1)$	$\delta(\text{in}k_i, d)\omega\left(\frac{1}{d}\right)$	$1 - A_{i,j}(t+1)$	$\delta(\text{in}k_i, d)\omega\left(\frac{1}{d}\right)$	$1 - A_{i,j}(t+1)$
1	0	0	0	1	0	1	0	1	0	1	0
1	0	0	1	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	1	1	0	$1 - A_{j,i}(t+1)$	$\delta(\text{in}k_i, d)\omega\left(\frac{1}{d}\right)$	$1 - A_{j,i}(t+1)$	$\delta(\text{in}k_i, d)\omega\left(\frac{1}{d}\right)$	$1 - A_{j,i}(t+1)$	$\delta(\text{in}k_i, d)\omega\left(\frac{1}{d}\right)$

TABLE I: Different rulesets (the columns labeled ‘time $t + 1$ ’) for three biologically inspired models: (a) the simplest scenario, (b) incorporating conservation of resources, and (c) implementing a delay factor (see text for details). As $A_{i,j}$ and $A_{j,i}$ are mutually exclusive, there are only 12 possible states of a link at time t .

We can then write the update for the resource distribution process as

$$\underline{S}(t+1) = \mathbf{T}^\dagger(t+1)\underline{S}(t) + \underline{\xi}(t), \quad (5)$$

where the vector $\underline{\xi}$ corresponds to the accumulation of resource by agents from the resource (substrate) layer. We impose the constraint that only ‘‘alive’’ (i.e. active) agents can accumulate resource through this process so

$$\xi_i(t) = R_E \phi(S_i(t)) L_i, \quad (6)$$

where the vector \underline{L} denotes the (binary) existence of resource at position of node (agent) i in the resource layer, R_E is the rate at which an agent accumulates the resource and $\phi(x)$ is the step function defined earlier. This resource accumulation from the substrate could be made time dependent (i.e. finite resources) although here we will not consider this effect. Equations 4, 5 and 6 constitute Process 1 and represent the functional update stage of Eq. 3.

For the example of Fig. 3, the amount of resources that node i has at time $t + 1$ is

$$\begin{aligned} S_i(t+1) &= S_j(t) \frac{A_{j,i}(t+1)}{\text{out}k_j(t+1)} + S_m(t) \frac{A_{m,i}(t+1)}{\text{out}k_m(t+1)} \\ &\quad + S_r(t) \frac{A_{r,i}(t+1)}{\text{out}k_r(t+1)} + S_q(t) \frac{A_{q,i}(t+1)}{\text{out}k_q(t+1)} \\ &= S_j(t) T_{j,i}(t+1) + S_r(t) T_{r,i}(t+1) \\ &\quad + S_q(t) T_{q,i}(t+1). \end{aligned} \quad (7)$$

We can now observe the Network Automaton in operation as shown in Fig. 4. We start with a single node η above a single food source so that at time $t = 0$ the agent has some resource. In the initial configuration the adjacency matrix is all zeros $A_{i,j}(0) = 0 \forall i, j$, and the resource information vector is all zeros except $S_\eta(0) = R_E$

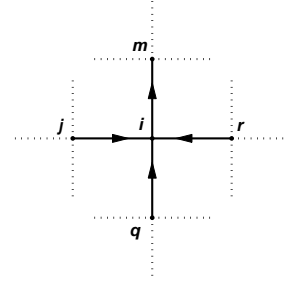


FIG. 3: The influx of resource into node labelled i . The amount this node receives from node j is related to the out-degree of node j . This amount would be expressed as $S_j(t)/\text{out}k_j(t+1)$.

such that the initial agent has amount R_E . For this example, the resource accumulation vector $\underline{\xi}(t)$ is also all zeros except $\xi_\eta(t) = R_E \forall t$. We observe both the network and functional aspect of the system. The nodes (agents) are superimposed on the directed network, and the amount of resources a node has, is indicated by its color ranging from blue (low concentration) to dark red (high concentration) [12, 16]. Only nodes that have resources are included and the result is independent of the choice of R_E . A longer simulation is shown in Fig. 5.

Note that under this ruleset and functional update stage an agent might accumulate resource indefinitely. By allowing a node which has in-degree equal to the maximum connectivity ($\text{in}k_i = d$) to randomly flip a direction of one of its links overcomes this biologically unfeasible behavior and results in the topological update of (Rule-set b) where the Kronecker delta function is defined as $\delta(x, y) = 0$ for $x \neq y$ and $\delta(x, y) = 1$ for $x = y$. A simulation of this ruleset coupled with the functional update of

Process 1 is shown in Fig. 5 and emergent canalized flux channels are clear. Such channels have been observed experimentally in a wide class of real biological fungi [15].

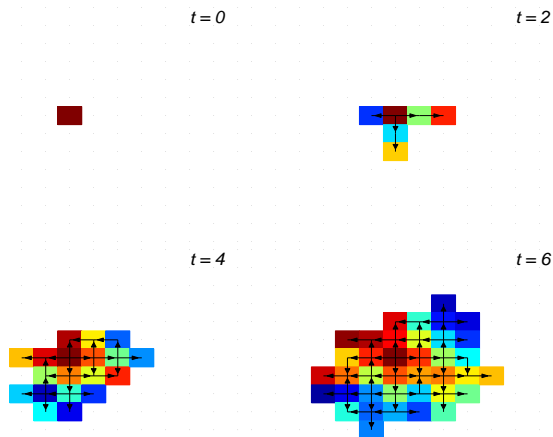


FIG. 4: (Color online) The evolution of the biologically inspired NA over 6 timesteps. The colors represent the amount of resource $S_i(t)$ a node has (the functional aspect) superimposed on a directed network (the structural aspect).

In Rulesets **a** and **b**, the physical transport of nutrient is of comparable speed to that of the growth, which is clearly not reasonable for most biological systems. A further development can be made to incorporate two different time scales in the model so that the growth and redistribution of resources take place at different rates, leading to Ruleset **c**. To establish this feature within the model, we can delay the growth process by introducing a parameter g to the stochastic growth terms in the ruleset, so that an agent grows on average one neighbor every $1/g$ time steps. This is depicted explicitly in Table I. The result of such a delay is the build up of resource towards the periphery of the object as one might expect [15]. This is illustrated in Fig. 1 for the real fungi and in Fig. 5 for the model.

We can also incorporate consumption of resources by agents in the model by making further modifications to the functional update stage, resulting in Process 2. If an agent has more than some residual consumption amount, R_C , then this rate of consumption is deducted. If the agent has less than this value, all of that agent's resources are removed and that agent might be considered dead. The initial stage in the functional update process can be described in a similar manner to Eq. 5:

$$\underline{S}'(t+1) = \mathbf{T}^\dagger(t+1)\underline{S}(t) + \underline{\xi}(t), \quad (8)$$

where the resource distribution is described by the transition matrix $\mathbf{T}(t+1)$ defined in Eq. 4 and the accumulation of resource from the substrate is described by the vector $\underline{\xi}(t)$ defined in Eq. 6. We can then write the update for

$\underline{S}(t+1)$, including consumption of resource as

$$S_i(t+1) = \phi(S'_i(t+1) - R_C)(S'_i(t+1) - R_C), \quad (9)$$

which makes use of the step function $\phi(x)$. Only agents (nodes) active in the network can accumulate resources from the resource layer. The effect of this “cost” of living clearly limits the potential size of the system. Equations 4, 6, 8 and 9 constitute Process 2 and the effect of using this functional update are illustrated in Fig. 5.

IV. CONCLUDING REMARKS AND DISCUSSION

In this paper we have developed the concepts of Network Automata (NA) which can couple evolution and function of complex networks by using simple microscopic rules at the level of nodes and links. We have demonstrated the practicability of the framework by applying it to a class of biologically inspired models, which produce qualitatively similar canalised flow patterns to those observed in real woodland fungi [15, 16, 17, 19]. This suggests that for organisms that have to adapt their morphology to a variable environment, function may play a crucial role in determining structure. The well defined and simple rulesets not only make replication straightforward but also aid implementation at the programming level. In the study of emergent phenomena involving networks, this framework enables concise and clear establishment of microscopic rules. One can easily think of more complex rulesets to more accurately model a real system, such as adding transport costs or finite resources, both of which can easily be accomplished, or even a time-dependent ruleset. It is then interesting to ask what level of complexity is required to more accurately model real biological systems.

We expect there to be many application domains for Network Automata from social to biological systems. One can envisage applying the NA framework also to discrete differential equation modeling and Diffusion Limited Aggregation (DLA) systems and, indeed, to any system in which the dynamics of network topology is related to the function performed thereon. There are certain biological systems which can quasi-solve increasingly complex problems in a constant time. An example is *Physarum polycephalum*, the true slime mold, which approximately identifies the Steiner points when placed upon multiple food sources [20]. Given that it might be possible to model these systems within the NA framework, it might suggest how to design a hardware based implementation to perform similar calculations in constant time. It is interesting to pose the question as to what kind of problems could be solved by such a system and how complex the microscopic rules would be for a given problem. This would reflect the minimum length of ruleset that would have to be employed by the system in both the network and functional update stages.

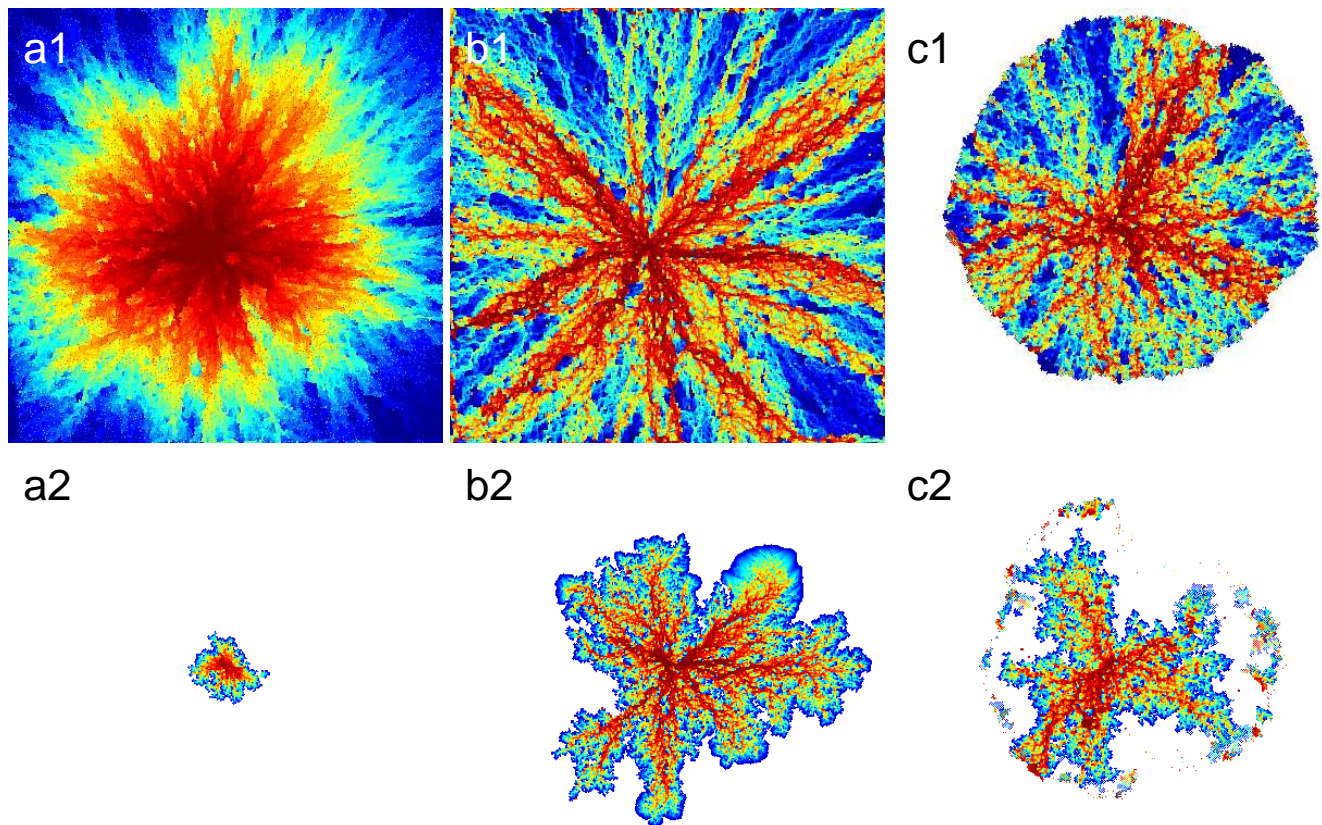


FIG. 5: (Color online) Simulated output for the Rulesets **a**, **b** and **c** and functional updates of Processes **1** and **2**. For all models, the lattice is of dimension 400×400 with periodic boundary conditions. Each was seeded with a single initial agent above a single (infinite) resource. The rate of resource accumulation for that agent is $R_E = 80,000$ and the simulation was run for 2000 time steps for each model. For Process **2** the residual consumption rate of each agent is set at $R_C = 1$. For Ruleset **c**, the delay factor is $g = 0.1$.

D.M.D.S. acknowledges funding by EPSRC and BB-SRC and J.-P.O. by a Wolfson College Junior Research Fellowship (Oxford). D.M.D.S acknowledges funding

from the European Union (MMCOMNET) for part of this research. We thank Felix Reed-Tsochas for comments and suggestions.

-
- [1] R. Albert and A.-L. Barabási, *Rev. Mod. Phys.* **74**, 47 (2002).
 - [2] M.E.J. Newman, *SIAM Rev.* **45**, 167 (2003).
 - [3] S.N. Dorogovtsev and J.F.F. Mendes, *Adv. Phys.* **51**, 1079 (2002).
 - [4] A.L. Barabási and R. Albert, *Science* **286**, 509 (1999).
 - [5] S. Wolfram, *A New Kind of Science* (Wolfram Media, 2002).
 - [6] A.E. Motter, M.A. Matias, J. Kurths, and E. Ott, *Physica D* **224**, vii-viii (2006).
 - [7] R. Lambiotte, J.-C. Delvenne, M. Barahona, arXiv:0812.1770 (2008).
 - [8] P. Fonczak, A. Fronczak and J.A. Hołyst, *Phys. Rev. E* **73**, 046117 (2006).
 - [9] M.G. Zimmermann, V.M. Eguilez and M. San Miguel, *Phys. Rev. E* **69**, 065102(R) (2004).
 - [10] C. Meisel and T. Gross, arXiv:0903.2987, (2009).
 - [11] T. Rohlf and S. Bornholdt, arXiv:0811.0980, (2008).
 - [12] D.M.D. Smith, *Agents, Games and Networks*, DPhil Thesis, University of Oxford, (2007).
 - [13] D. Bebbler, J. Hynes, P.R. Darrah, L. Boddy and M.D. Fricker, *Proc. Roy. Soc. B* **274** 2307-2315, (2007).
 - [14] M. Tlalka, S.C. Watkinson, P.R. Darrah and M.D. Fricker, *New Phytologist* **153**, 173184 (2002).
 - [15] M. Tlalka, D. Hensman, P.R. Darrah, S.C. Watkinson, and M.D. Fricker, *New Phytologist* **158**, 325 (2003).
 - [16] M.D. Fricker, J.A. Lee, D.P. Bebbler, M. Tlalka, J. Hynes, P.R. Darrah, S.C. Watkinson and L. Boddy, *J. Microscopy* **231**, 317-331, (2008).
 - [17] M. Tlalka, D. Bebbler, P.R. Darrah, S.C. Watkinson and M.D. Fricker, *Fungal Genetics and Biology* **44** 1085-1095, (2007).

- [18] J. von Neumann and A. W. Burks, *Theory of Self-reproducing Automata* (University of Illinois Press, Urbana, 1966).
- [19] M. Tlalka, D.P. Bebbler, P.R. Darrah, S.C. Watkinson and M.D. Fricker, *Fungal Genetics and Biology* **45** 1111-1121, (2008).
- [20] T. Nakagaki, R. Kobayashi, Y. Nishiura, and T. Ueda, *Proc. Biol. Sci.* **271**, 1554:2305 (2004).
- [21] Because each link can see the requisite information at the start of the time step, the system is synchronously updated. This implementation could then encompass asynchronous updates through application over longer timescales.
- [22] The concept of a Network Automaton has been alluded to before although not formally defined [5].
- [23] The discrete time recurrence relation is more appropriate than the continuous time coupled differential equation analogy $\frac{d\mathbf{A}}{dt} = f(\mathbf{A}, \mathbf{S})$, $\frac{d\mathbf{S}}{dt} = g(\mathbf{A}, \mathbf{S})$ in that it allows discontinuous changes in the system's update.